Discrete Mathematics 344 (2021) 112316

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

Leprechauns on the chessboard

Guillaume Escamocher, Barry O'Sullivan

Insight Centre for Data Analytics, School of Computer Science & Information Technology, University College Cork, Ireland

ARTICLE INFO

Article history:
Received 14 August 2020
Received in revised form 10 December 2020
Accepted 15 January 2021
Available online xxxx

Keywords:
Constraint satisfaction
n-Queens Problem
Assignment diversity

ABSTRACT

We introduce in this paper leprechauns, fairy chess pieces that can move either like the standard queen, or to any tile within k king moves. We then study the problem of placing n leprechauns on an $n \times n$ chessboard. When $k = 1$, this is equivalent to the standard n-Queens Problem. We solve the problem for $k = 2$, as well as for $k > 2$ and $n \leq (k+1)^2$, giving in the process an upper bound on the lowest non-trivial value of n such that the (k, n)-Leprechauns Problem is satisfiable. Solving this problem for all k would be equivalent to solving the diverse n-Queens Problem, the variant of the n-Queens Problem where the distance between the two closest queens is maximized. While diversity has been a popular topic in other constraint problems, this is not the case for the n-Queens Problem, making our results the first major ones in the domain.

1. Introduction

The n-Queens Problem asks how to place n chess pieces on an $n \times n$ board such that no two pieces share the same row, column or diagonal. Finding such an arrangement can be done in linear time for every n apart from $n = 2$ and $n = 3$, for which none exist [19,20]. Despite its relative easiness, the n-Queens Problem has proven to be very popular during the last one and a half centuries, giving birth to several related combinatorial problems. Some of these variants remain tractable, like the modular n-Queens Problem, where opposite sides of the board are merged to form a torus [22]. Some of them on the other hand are NP-Complete, like the n-Queens Completion Problem which asks whether a given set of queens that have already been placed on the board can be extended to a solution to the standard n-Queens Problem [9]. The complexity of others is still open, see for example the question of finding the lexicographically minimal solution to the n-Queens Problem [8].

An intuitive n-Queens Problem variation that, to the best of our knowledge, has never been studied, is the Diverse n-Queens Problem. In this problem, the goal is to find a solution that not only fulfills the original constraints, but also places the queens as far away as possible from each other. The absence of work on the subject is regrettable, because diversity is a popular topic in constraint programming [1,4,14,15,21], with numerous applications that include recommender systems [27], exact satisfiability [5], car configuration [13,16], and architectural tests [17].

Finding a solution to the Diverse n-Queens Problem is equivalent to finding a solution to the standard n-Queens Problem where all queens are more than k away from each other, with k being a distance parameter corresponding to the number of moves a king needs to move between two tiles. Indeed, in order to find a solution for the former, one can simply solve the latter for k from 1 to $n-1$ and keep the largest k for which a solution was found. In the other direction, whether a solution to the Diverse n-Queens Problem has all queens more than k apart from each other determines if there is a solution to the n-Queens problem with the additional distance constraint.

E-mail addresses: guillaume.escamocher@insight-centre.org (G. Escamocher), b.osullivan@cs.ucc.ie (B. O'Sullivan).

https://doi.org/10.1016/j.disc.2021.112316

To model the requirement that all queens be more than k away from each other, we use a fairy[1] chess piece that we name a *leprechaun*. Leprechauns can move either like a queen, or to any tile of the board that is at most k rows and at most k columns away from them, with k being their *range*. The problem of placing n queens on an $n \times n$ board such that each one of them is (strictly) more than k away from her nearest neighbor becomes then the problem of placing n non-attacking range-k leprechauns on that same $n \times n$ board. We call this problem the (k, n)-Leprechauns Problem. The distance parameter k establishes a hierarchy over (k, n)-Leprechauns Problems: the higher k is, the more tiles a range-k leprechaun can potentially move to, and therefore the harder it is to find a configuration of n non-attacking leprechauns.

In this paper, we take the first steps towards a characterization of the Leprechauns Problem depending on the parameters k and n. As we mention among the definitions in the next section, a range-1 leprechaun is a queen, so the $(1, n)$-Leprechauns Problem is already solved since it corresponds to the standard n-Queens problem. In Section 3, we present the first of our two main results: a construction of a solution to the $(2, n)$-Leprechauns Problem for all $n \geq 10$. While the (un-)satisfiability of this particular problem was already known for $n < 10$ [10], our algorithm is the first, as far as we know, to completely solve the $(2, n)$-Leprechauns Problem for any value of n.

For higher values of k, the (k, n)-Leprechauns Problem appears to be more challenging. We suspect, however, that the frontier between unsatisfiability and satisfiability lies around the $n = (k+1)^2$ parabola. Indeed, as we show in Section 4, the (k, n)-Leprechauns Problem is always unsatisfiable (except for the trivial case $n = 1$) if $n < (k+1)^2$, but can go either way if $n = (k+1)^2$, where the existence of a solution depends on the parity of k. Most of the section is devoted to the proof of this result, while the remainder presents miscellaneous empirical observations. We conclude in Section 5.

2. Definitions

A (chess)*board* is a grid of dimensions $n \times n$ containing n^2 *tiles*. Each tile has coordinates (i, j), with i between 1 and n indicating the column and j between 1 and n indicating the row. We use the standard convention of numbering the columns from left to right and the rows from bottom to top. Diagonals can be labeled as well.

Definition 1 (*Diagonals*)**.** The *sum diagonal* d_+ is the set of tiles $\{(i, j) | i + j = d_+\}$. The *difference diagonal* d_- is the set of tiles $\{(i, j) | i - j = d_-\}$.

To define our distance metric, we use the appropriately named chessboard distance:

Definition 2 (*Chessboard Distance*)**.** Let t_1 and t_2 be two tiles of respective coordinates (i_1, j_1) and (i_2, j_2). We say that the *chessboard distance* between t_1 and t_2 is the maximum between $|i_1 - i_2|$ and $|j_1 - j_2|$.

The chessboard distance is sometimes called *Chebyshev distance*. Since this is the only distance metric we will be using, we will from now on simply refer to it as the *distance*.

The game of chess uses figurines, or pieces, that can moved from one tile to another. The n-Queens Problem focuses on the eponymous queen piece:

Definition 3 (*Queen*)**.** A *queen* is a chess piece that can move from a tile (i, j) to any other tile (i', j') as long as one of the following is true:

- $i' = i$
- $j' = j$
- $i' + j' = i + j$
- $i' - j' = i - j$

The first condition corresponds to a move on the same column, the second to a move on the same row. The last two conditions correspond to a diagonal move. The queen is a part of the standard chess figurine set. For recreational purposes, many non-standard chess pieces have been invented. We now introduce the leprechaun chess piece:

Definition 4 (*Range-k Leprechaun*)**.** A *range-k leprechaun* is a chess piece that can move from a tile t_1 to any other tile t_2 as long as one of the two following conditions is true:

- t_2 can be reached from t_1 by a queen move.
- The distance between t_2 and t_1 is at most k.

A range-1 leprechaun is a queen. A range-2 leprechaun is an amazon, or superqueen, although it has been called many names throughout history [3], like for example maharajah [7], and was even used as a substitute for the queen in ancient Russian chess [12]. A range-3 leprechaun can be seen as the combination of a queen, a knight, a camel and a zebra, where camels and zebras are fairy chess pieces that move to a tile (1,3) and (2,3) away respectively, leaping over intermediary pieces.

[1] Fairy chess pieces are chess pieces that are not part of the standard chess set.

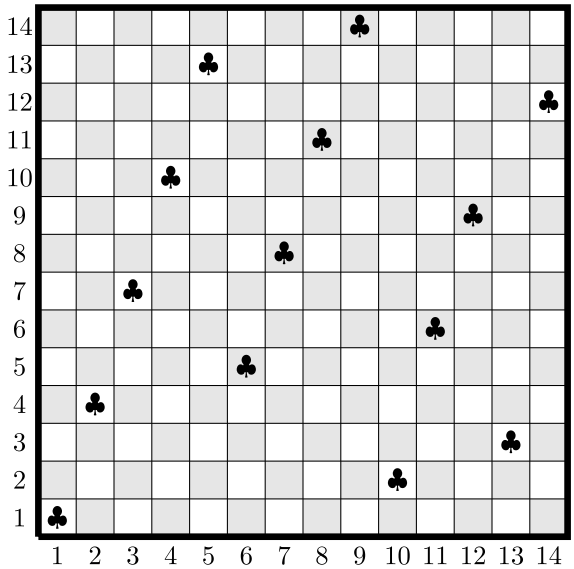

Fig. 1. A solution to the (2, 14)-Leprechauns Problem.

Definition 5 (*Attack*)**.** Let t be a tile of the board. We say that a chess piece *attacks* t if it can move from its current tile to t in one move. We say that a chess piece *attacks* another chess piece if the latter is placed on a tile attacked by the former.

The n-Queens Problem can be generalized to numerous other (fairy) chess pieces, and the leprechaun is no exception.

Definition 6 (*(k, n)-Leprechauns Problem*)**.** The *(k, n)-Leprechauns Problem* is to find a way to place n non-attacking range-k leprechauns on an $n \times n$ board, or to prove that there is no such arrangement.

A solution to the (k, n)-Leprechauns Problem will be written as an ordered list of n rows, indexed by columns. In other words, if the ith element of a solution is j, then this solution contains a leprechaun on tile (i, j).

The board pictured in Fig. 1 illustrates the notions defined in this section. Its dimensions are 14×14, and it contains 14 leprechauns, each represented by a shamrock. The leprechaun in tile $(11, 6)$ is in Sum Diagonal 17, because its coordinates add to 17, and in Difference Diagonal 5, because the row component of its coordinates subtracted from the column component gives 5. It is the only leprechaun in these diagonals, which go from $(3, 14)$ to $(14, 3)$ and from $(6, 1)$ to $(14, 9)$, respectively, and is also alone in Row 6 and Column 11. Its nearest neighbors are the leprechauns in tiles (13,3) and (12,9), each at distance 3, so if it is a range-1 or range-2 leprechaun, then it does not attack them. In fact, this board is a solution for the $(1, n)$- and $(2, n)$-Leprechauns Problems, which can be written as $(1, 4, 7, 10, 13, 5, 8, 11, 14, 2, 6, 9, 3, 12)$. However this board is not a solution to the (k, n)-Leprechauns Problem for $k \geq 3$, because several pairs of leprechauns are within distance 3 of each other.

To remain consistent with the labeling of the board axes, we will use "mȯdulo" and $m\dot{o}d$ instead of "modulo" and *mod*, where $a \; m\dot{o}d \; b$ is equal to b if a is divisible by b, and to $a \bmod b$ otherwise. So for example $28 \; m\dot{o}d \; 14 = 14$.

3. The (2, *n*)-Leprechauns problem

The case of the range-1 leprechaun is equivalent to the n-Queens Problem, for which a linear-time algorithm to find a solution for each n (apart from $n = 2$ and $n = 3$ for which there is no solution) has been known since the nineteenth century [19,20]. In contrast, we are not aware of any algorithm, regardless of complexity, that can solve the $(2, n)$-Leprechauns Problem for any n.

For the first few values of n, the number of solutions to the $(2, n)$-Leprechauns Problem is given at the On-line Encyclopedia of Integer Sequences [25], where $(2, n)$-Leprechauns are called "superqueens". In 2004 a mention was made on that web-site of code by Frank Schwellinger that could solve the problem in linear time, but the link is no longer active. A cached version[2] reveals his algorithm, which appears to be incorrect when n is congruent to 2 or to 3 mȯdulo 6 and $n \geq 26$. For $n = 50$ for example, his construction places superqueens on tiles (11,44) and (36,19), which belong to the same diagonal. A subsequent literature review of the problem [2] does not mention any algorithm that works for all n.

2 https://web.archive.org/web/20040823045854/http://super.info.ms/.

The only results so far for the $(2, n)$-Leprechauns Problem are partial. If $n \geq 10$ is a prime or 1 less than a prime, then there is a solution to the $(2, k)$-Leprechauns Problem [11]. If $n < 10$, then there is a solution to the $(2, n)$-Leprechauns Problem if and only if $n = 1$ [10]. The problem has also been solved when the number of leprechauns to place is strictly less than n and the board is a torus [18].

Many solutions to the n-Queens Problem are based on linear congruence algorithms, meaning that the queen in each column is a constant number of rows (mȯdulo n) above the queen from the previous column. Because of arithmetical properties, these methods have to vary the exact height of this step, and possibly make other adjustments to the construction, depending on the value of n mȯdulo 6. When a step of at least $k + 1$ rows is allowed between queens from consecutive columns, the additional distance constraint inherent to range-k leprechauns will be fulfilled and the n-queens solutions can be used as is for the (k, n)-Leprechaun Problem. Unfortunately, in some cases (most often when $n \equiv 2 \; \dot{mod}\; 6$ or $n \equiv 3 \; \dot{mod}\; 6$) strict conditions on the structure of n-Queens solution do not allow for the leprechaun distance constraints to be satisfied without substantial alterations to the algorithm itself.

We are now going to present a linear-time algorithm that for any n either returns a solution to the $(2, n)$-Leprechauns Problem, or indicates that none exists. Our algorithm also distinguishes cases depending on the congruence of n mȯdulo 6, although the particular case of $n \equiv 2 \; \dot{mod}\; 6$ will be further partitioned, with its two main subcases being $n \equiv 2 \; \dot{mod}\; 12$ and $n \equiv 8 \; \dot{mod}\; 12$. For $n \equiv 1 \; \dot{mod}\; 6$, $n \equiv 3 \; \dot{mod}\; 6$, and $n \equiv 5 \; \dot{mod}\; 6$, we will use one of the n-Queens constructions that allow for jumps of 3 rows from one column to the next [6]. We start with the cases $1 \; \dot{mod}\; 6$ and $5 \; \dot{mod}\; 6$, which are treated in Algorithm 1.

Data: An integer n such that $n \equiv 1 \; \dot{mod}\; 6$ or $n \equiv 5 \; \dot{mod}\; 6$, and $n = 1$ or $n \geq 11$.
Result: A solution to the $(2, n)$-Leprechauns Problem.
1 Declare Sol, an array with n cells;
2 $Sol[1] \leftarrow 1$;
3 **for** $c \leftarrow 2$ **to** n **do** $Sol[c] \leftarrow (Sol[c-1] + 3) \; \dot{mod}\; n$;
4 **return** Sol;

Algorithm 1: When $n \equiv 1 \; \dot{mod}\; 6$ or $n \equiv 5 \; \dot{mod}\; 6$.

Lemma 1. *Let n be an integer such that $n \equiv 1 \; \dot{mod}\; 6$ or $n \equiv 5 \; \dot{mod}\; 6$, and such that $n = 1$ or $n \geq 11$. Then Algorithm 1 returns a solution to the $(2, n)$-Leprechauns Problem.*

Proof. If $n = 1$, then the solution consists of a single leprechaun and the satisfiability is trivial. Otherwise, we have $n \geq 11$. Suppose that the leprechauns in tiles (i, j) and (i', j') attack each other. Assume without loss of generality that $i < i'$. We know that the algorithm gives a solution to the n-Queens Problem [6], so the only way for these leprechauns to attack each other is if they are within distance 2 of each other. So $i' = i + 1$ or $i' = i + 2$. So $j' \; \dot{mod}\; n = j + 3 \; \dot{mod}\; n$ or $j' \; \dot{mod}\; n = j+6 \; \dot{mod}\; n$. The two rows above Row j are Rows j_1 and j_2 with $j_1 \; \dot{mod}\; n = j+1 \; \dot{mod}\; n$ and $j_2 \; \dot{mod}\; n = j+2 \; \dot{mod}\; n$. The two rows below Row j are Rows j_{-1} and j_{-2} with $j_{-1} \; \dot{mod}\; n = j + n - 1 \; \dot{mod}\; n$ and $j_{-2} \; \dot{mod}\; n = j + n - 2 \; \dot{mod}\; n$. Since $n \geq 11$, Row j' is neither one of the two rows above Row j nor one of the two rows below Row j. Since the queens constraints are satisfied, we also know that Row j' is not Row j. So the leprechauns on tiles (i, j) and (i', j') cannot be within distance 2 of each other. This proves the validity of the algorithm. □

Fig. 2 illustrates the solution output by Algorithm 1 for $n = 13$. Notice that the algorithm places a leprechaun on the bottom left tile of the board (Line 2). This means that by cropping out the first row and the first column of the $n \times n$ board, we get an arrangement of $n-1$ non-attacking leprechauns on an $(n-1) \times (n-1)$ board, which is a solution to the $(2, n-1)$-Leprechauns Problem. For example, the top right part of the board in Fig. 2 is a solution to the $(2, 12)$-Leprechauns Problem. Algorithm 2 exploits the property to solve the cases $n \equiv 4 \; \dot{mod}\; 6$ and $n \equiv 6 \; \dot{mod}\; 6$.

Data: An integer $n \geq 10$ such that $n \equiv 4 \; \dot{mod}\; 6$ or $n \equiv 6 \; \dot{mod}\; 6$.
Result: A solution to the $(2, n)$-Leprechauns Problem.
1 Declare Sol, an array with n cells;
2 $Sol[1] \leftarrow 3$;
3 **for** $c \leftarrow 2$ **to** n **do** $Sol[c] \leftarrow (Sol[c-1] + 3) \; \dot{mod}\; (n+1)$;
4 **return** Sol;

Algorithm 2: When $n \equiv 4 \; \dot{mod}\; 6$ or $n \equiv 6 \; \dot{mod}\; 6$.

Lemma 2. *Let $n \geq 10$ be an integer such that $n \equiv 4 \; \dot{mod}\; 6$ or $n \equiv 6 \; \dot{mod}\; 6$. Then Algorithm 2 returns a solution to the $(2, n)$-Leprechauns Problem.*

Proof. We just need to prove that Algorithm 2 returns the same board as Algorithm 1, minus the bottom row r_{bottom} and the leftmost column c_{left}. Observe that the congruence is modulo $n+1$, which compensates for the lack of the bottom row r_{bottom} when $Sol[c-1] + 3$ is greater than n. So the only thing to check is that the result of the congruence mȯdulo $n+1$ is never $n+1$. We know that $n+1$ is congruent to either 1 or 5 mȯdulo 6, so it is prime with 3. Furthermore, the value in the first cell of the solution is 3 (Line 2), and the height of the jump between consecutive columns is also 3 (Line 3),

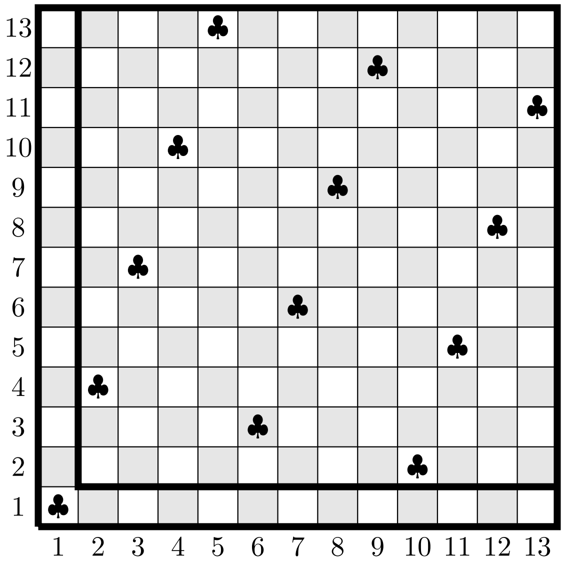

Fig. 2. A solution to the (2, 13)-Leprechauns Problem.

so the first time that the result of the congruence will be $n+1$ is at the $(n+1)$th column, which is not used because of the lack of the leftmost column c_{left}. □

For $n \equiv 3 \; m\dot{o}d \; 6$, we use one of the rare constructions that allow a jump height of 3 for this particular case [6]. The method is presented in Algorithm 3, and illustrated in Fig. 3.

Data: An integer $n \geq 15$ such that $n \equiv 3 \; m\dot{o}d \; 6$.
Result: A solution to the $(2, n)$-Leprechauns Problem.
1 Declare *Sol*, an array with n cells;
2 $Sol[1] \leftarrow 5$; // Can actually be any value between 5 and $n-9$ inclusive.
3 **for** $c \leftarrow 2$ **to** $\frac{n}{3}$ **do**
4 | $Sol[c] \leftarrow (Sol[c-1]+3) \; m\dot{o}d \; n$;
5 **end**
6 $Sol[\frac{n}{3}+1] \leftarrow Sol[\frac{n}{3}]+7$;
7 **for** $c \leftarrow \frac{n}{3}+2$ **to** $\frac{2n}{3}$ **do**
8 | $Sol[c] \leftarrow (Sol[c-1]+3) \; m\dot{o}d \; n$;
9 **end**
10 $Sol[\frac{2n}{3}+1] \leftarrow Sol[\frac{2n}{3}]+7$;
11 **for** $c \leftarrow \frac{2n}{3}+2$ **to** n **do**
12 | $Sol[c] \leftarrow (Sol[c-1]+3) \; m\dot{o}d \; n$;
13 **end**
14 **return** *Sol*;

Algorithm 3: When $n \equiv 3 \; m\dot{o}d \; 6$.

Lemma 3. *Let $n \geq 15$ be an integer such that $n \equiv 3 \; m\dot{o}d \; 6$. Then Algorithm 3 returns a solution to the $(2, n)$-Leprechauns Problem.*

Proof. The reasoning is similar to the one used in the proof of Lemma 1, in that we already know that the algorithm outputs a solution for the n-Queens Problem [6], so we only need to check that the distance constraints are satisfied.

Suppose that the leprechauns in tiles (i, j) and (i', j') attack each other. Assume without loss of generality that $i < i'$. Since queens constraints are fulfilled [6], the only way for these leprechauns to attack each other is if they are within distance 2 of each other. So $i' = i+1$ or $i' = i+2$. The height of the jump between successive columns is either 3 (Lines 4, 8, and 12) or 7 (Lines 6 and 10), and the latter does not occur consecutively (because it occurs at Columns $\frac{n}{3}+1$ and $\frac{2n}{3}+1$, and since $n \geq 15$ we have $\frac{n}{3}+1 < \frac{2n}{3}+1-1$), so j' can only be equal to $j+3 \; m\dot{o}d \; n$, $j+7 \; m\dot{o}d \; n$, or $j+10 \; m\dot{o}d \; n$. The two rows above Row j are Rows j_1 and j_2 with $j_1 \; m\dot{o}d \; n = j+1 \; m\dot{o}d \; n$ and $j_2 \; m\dot{o}d \; n = j+2 \; m\dot{o}d \; n$. The two rows below Row j are Rows j_{-1} and j_{-2} with $j_{-1} \; m\dot{o}d \; n = j+n-1 \; m\dot{o}d \; n$ and $j_{-2} \; m\dot{o}d \; n = j+n-2 \; m\dot{o}d \; n$. Since $n \geq 15$,

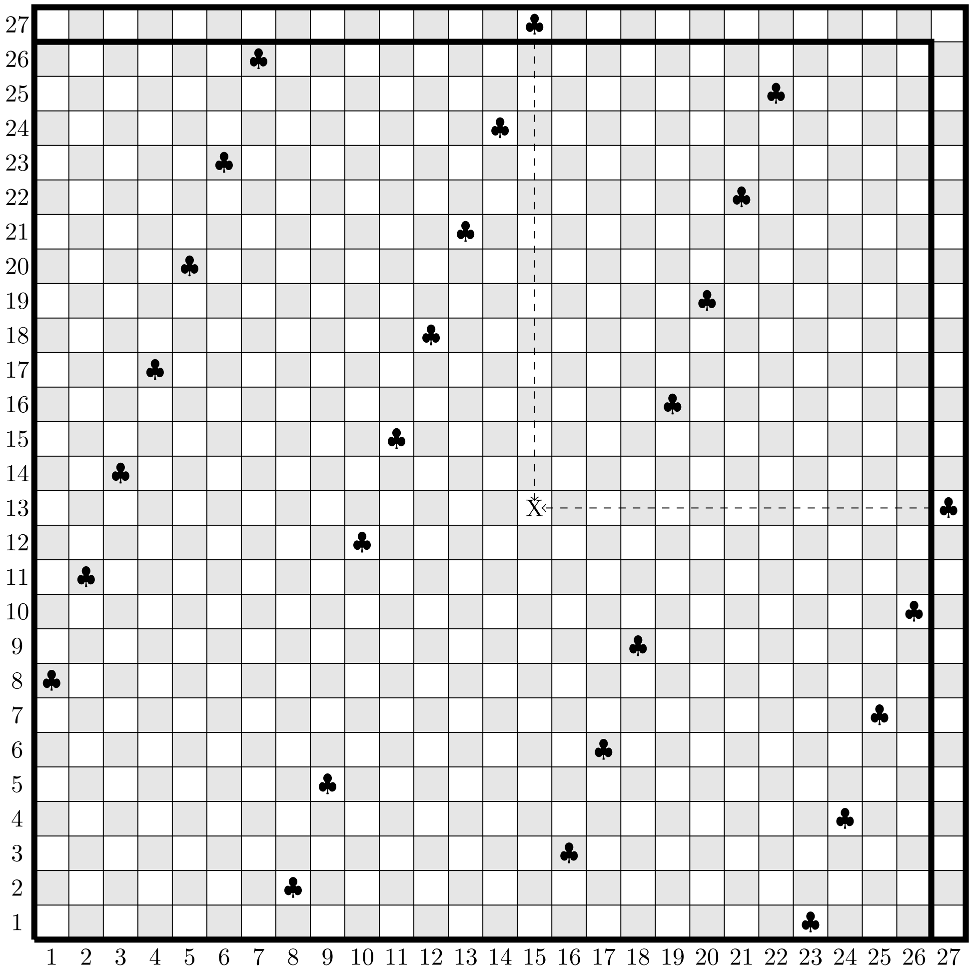

Fig. 3. A solution to the $(2, 27)$-Leprechauns Problem.

Row j' cannot be Row j, cannot be one of the two rows above Row j, and cannot be one of the two rows below Row j. So the leprechauns on tiles (i, j) and (i', j') cannot be within 2 of each other. This proves the validity of the algorithm. $\square$

Algorithm 3 places the first column's leprechaun in Row 5. Any row between Row 5 and Row $n - 9$ inclusive would have worked too [6]. For example, Fig. 3 illustrates for $n = 27$ the output of Algorithm 3 when placing the first column's leprechaun in Row 8 instead of Row 5. The result is still a solution to the $(2, 27)$-Leprechauns Problem. Furthermore, we never mention Line 2 in the proof of Lemma 3, so its correctness stands regardless of which row in this set is chosen.

On the other hand, placing the first column's leprechaun in either one of the first four rows or one of the last nine ones would lead to violated diagonal constraints if following the rest of Algorithm 3. In particular, we cannot use Algorithm 2's trick of placing a leprechaun in a corner of the board and then cropping one side row and one side column to get a solution to the $(2, n - 1)$-Leprechauns Problem.

When $n \equiv 2 \ m\dot{o}d\ 6$, applying Algorithm 3 to $n + 1$ and removing the last column of the output gives an arrangement of n non-attacking leprechauns on an $(n + 1) \times n$ board. Moving the last row's leprechaun to the one row among the first n that is empty transforms this into an arrangement of n leprechauns on an $n \times n$ board that respects row and column constraints, however other constraints might not be satisfied. As an example, cropping out the last column in Fig. 3 and moving the leprechaun in Column 15 from the last row to the now empty Row 13 places this leprechaun on the same diagonal as the one on tile (24,4). To obtain a general solution for the case $n \equiv 2 \ m\dot{o}d\ 6$ with this method, special care has to be taken when choosing in which row to place the first column's leprechaun. No row works for all n, but when $n \geq 20$ it is possible for all but six small values of n to determine an acceptable row for the first leprechaun by looking at the remainder of n divided by 12, as is done in Algorithm 4. Even the other six values of n can be treated with the same algorithm, by using their own separate formula to find out in which row to place the first column's leprechaun.

Data: An integer $n \geq 20$ such that $n \equiv 2 \ \dot{m}od\ 6$.
Result: A solution to the $(2, n)$-Leprechauns Problem.

1 Declare *Sol*, an array with n cells;
2 **if** $n = 20$ **or** $n = 26$ **or** $n = 32$ **or** $n = 38$ **or** $n = 44$ **or** $n = 56$ **then**
3 | $Sol[1] \leftarrow n - 10$;
4 **end**
5 **if** $n \geq 50$ **and** $n \equiv 2 \ \dot{m}od\ 12$ **then** $Sol[1] \leftarrow 8$;
6 **if** $n \geq 68$ **and** $n \equiv 8 \ \dot{m}od\ 12$ **then** $Sol[1] \leftarrow 14$;
7 **for** $c \leftarrow 2$ **to** $\frac{n+1}{3}$ **do**
8 | $Sol[c] \leftarrow (Sol[c-1] + 3) \ \dot{m}od\ (n+1)$;
9 **end**
10 $Sol[\frac{n+1}{3} + 1] \leftarrow Sol[\frac{n+1}{3}] + 7$;
11 **for** $c \leftarrow \frac{n+1}{3} + 2$ **to** $\frac{2(n+1)}{3}$ **do**
12 | $Sol[c] \leftarrow (Sol[c-1] + 3) \ \dot{m}od\ (n+1)$;
13 **end**
14 $Sol[\frac{2(n+1)}{3} + 1] \leftarrow Sol[\frac{2(n+1)}{3}] + 7$;
15 **for** $c \leftarrow \frac{2(n+1)}{3} + 2$ **to** n **do**
16 | $Sol[c] \leftarrow (Sol[c-1] + 3) \ \dot{m}od\ (n+1)$;
17 **end**
18 **if** $n = 20$ **or** $n = 26$ **or** $n = 32$ **or** $n = 38$ **or** $n = 44$ **or** $n = 56$ **then**
19 | $Sol[\frac{2(n+1)}{3} + 2] \leftarrow n - 5$;
20 **end**
21 **if** $n \geq 50$ **and** $n \equiv 2 \ \dot{m}od\ 12$ **then** $Sol[\frac{2(n+1)}{3} - 3] \leftarrow 13$;
22 **if** $n \geq 68$ **and** $n \equiv 8 \ \dot{m}od\ 12$ **then** $Sol[\frac{2(n+1)}{3} - 5] \leftarrow 19$;
23 **return** *Sol*;

Algorithm 4: When $n \equiv 2 \ \dot{m}od\ 6$.

Lemma 4. *Let $n \geq 20$ be an integer such that $n \equiv 2 \ \dot{m}od\ 6$. Then Algorithm 4 returns a solution to the $(2, n)$-Leprechauns Problem.*

Proof. If $n \in \{20, 26, 32, 38, 44, 56\}$, the outputs are as follows:

$n = 20$: (10,13,16,19,1,4,7,14,17,20,2,5,8,11,18,***15***,3,6,9,12)
$n = 26$: (16,19,22,25,1,4,7,10,13,20,23,26,2,5,8,11,14,17,24,***21***,3,6,9,12,15,18)
$n = 32$: (22,25,28,31,1,4,7,10,13,16,19,26,29,32,2,5,8,11,14,17,20,23,30,***27***,3,6,9,12,15,18,21,24)
$n = 38$: (28,31,34,37,1,4,7,10,13,16,19,22,25,32,35,38,2,5,8,11,14,17,20,23,26,29, 36,***33***,3,6,9,12,15,18,21,24,27,30)
$n = 44$: (34,37,40,43,1,4,7,10,13,16,19,22,25,28,31,38,41,44,2,5,8,11,14,17,20,23,26, 29,32,35,42,***39***,3,6,9,12,15,18,21,24,27,30,33,36)
$n = 56$: (46,49,52,55,1,4,7,10,13,16,19,22,25,28,31,34,37,40,43,50,53,56,2,5,8,11,14, 17,20,23,26,29,32,35,38,41,44,47,54,***51***,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48)

In each case, the bold value indicates the leprechaun that was moved in Lines 18–20. All of these outputs have been confirmed to be solutions by a constraint solver, so we will now assume that $n \geq 50$ and $n \equiv 2 \ \dot{m}od\ 12$ (respectively $n \geq 68$ and $n \equiv 8 \ \dot{m}od\ 12$).

Let us put aside for the moment the last three lines of the algorithm, and let us attempt to describe the state of the board after Line 17. The board can be partitioned in three. The first part covers Columns 1 to $\frac{n+1}{3}$ and is filled in Lines 5 to 9. The second part covers Columns $\frac{n+1}{3} + 1$ to $\frac{2(n+1)}{3}$ and is filled in Lines 10 to 13. Finally the last part covers Columns $\frac{2(n+1)}{3} + 1$ to n and is filled in Lines 14 to 17. In each part, the row of a leprechaun is determined by adding 3 ṁodulo $n+1$ to the row of the previous column's leprechaun. Since $n \equiv 2 \ \dot{m}od\ 6$, $n+1$ is divisible by 3, so all leprechauns in the same part are congruent to the same value ṁodulo 3. From Line 5 (respectively 6), we know that the first leprechaun is placed in Row 8 (respectively 14), so the rows of the leprechauns in the leftmost part of the board comprise all the numbers less than $n+1$ that are equal to 2 ṁodulo 3, in the following order: $[8, 11, \ldots, n-3, n, 2, 5]$ (respectively $[14, 17, \ldots, n-3, n, 2, 5, 8, 11]$). From that and Line 10, we know that the next leprechaun is placed in Row 12 (respectively 18), so the rows of the leprechauns in the second part of the board comprise all the numbers less or equal than $n+1$ that are equal to 3 ṁodulo 3, in the following order: $[12, 15, \ldots, n-5, n-2, n+1, 3, 6, 9]$ (respectively $[18, 21, \ldots, n-5, n-2, n+1, 3, 6, 9, 12, 15]$). With Line 14 we now know that the next leprechaun is placed in Row 16 (respectively 22), so the rows of the leprechauns in the rightmost part of the board comprise all the numbers less than $n+1$ that are equal to 1 ṁodulo 3, except the number that corresponds to the row of the leprechaun that would

Table 1
Leprechaun coordinates when $n \geq 50$ and $n \equiv 2 \ \dot{mod}\ 12$.

Column c	Row	Sum diagonal	Difference diagonal
1 to $\frac{n+1}{3} - 2$	$8 + 3(c-1)$	$4c + 5$	$-2c - 5$
$\frac{n+1}{3} - 1$ to $\frac{n+1}{3}$	2, 5	$\frac{n+1}{3} + (1, 5)$	$\frac{n+1}{3} - (3, 5)$
$\frac{n+1}{3} + 1$ to $\frac{2(n+1)}{3} - 3$	$12 + 3(c - \frac{n+1}{3} - 1)$	$4c + 8 - n$	$-2c - 8 + n$
$\frac{2(n+1)}{3} - 2$ to $\frac{2(n+1)}{3}$	3, 6, 9	$\frac{2(n+1)}{3} + (1, 5, 9)$	$\frac{2(n+1)}{3} - (5, 7, 9)$
$\frac{2(n+1)}{3} + 1$ to $n - 4$	$16 + 3(c - \frac{2(n+1)}{3} - 1)$	$4c + 11 - 2n$	$-2c - 11 + 2n$
$n - 3$ to n	1, 4, 7, 10	$n + (-2, 2, 6, 10)$	$n - (4, 6, 8, 10)$

Table 2
Leprechaun coordinates when $n \geq 68$ and $n \equiv 8 \ \dot{mod}\ 12$.

Column c	Row	Sum diagonal	Difference diagonal
1 to $\frac{n+1}{3} - 4$	$14 + 3(c-1)$	$4c + 11$	$-2c - 11$
$\frac{n+1}{3} - 3$ to $\frac{n+1}{3}$	2, 5, 8, 11	$\frac{n+1}{3} + (-1, 3, 7, 11)$	$\frac{n+1}{3} - (5, 7, 9, 11)$
$\frac{n+1}{3} + 1$ to $\frac{2(n+1)}{3} - 5$	$18 + 3(c - \frac{n+1}{3} - 1)$	$4c + 14 - n$	$-2c - 14 + n$
$\frac{2(n+1)}{3} - 4$ to $\frac{2(n+1)}{3}$	3, 6, 9, 12, 15	$\frac{2(n+1)}{3} + (-1, 3, 7, 11, 15)$	$\frac{2(n+1)}{3} - (7, 9, 11, 13, 15)$
$\frac{2(n+1)}{3} + 1$ to $n - 6$	$22 + 3(c - \frac{2(n+1)}{3} - 1)$	$4c + 17 - 2n$	$-2c - 17 + 2n$
$n - 5$ to n	1, 4, 7, 10, 13, 16	$n + (-4, 0, 4, 8, 12, 16)$	$n - (6, 8, 10, 12, 14, 16)$

have been placed in Column $n+1$ if the loop had iterated once more. The order of the rows in the last part is as follows: $[16, 19, \ldots, n-4, n-1, 1, 4, 7, 10]$ (respectively $[22, 25, \ldots, n-4, n-1, 1, 4, 7, 10, 13, 16]$), and all numbers less than $n+1$ have been placed in the solution, with the exception of 13 (respectively 19), that would have been placed in cell $n+1$. The formulas describing the coordinates of the leprechauns depending on which part of the board they belong are detailed in Table 1 for $n \geq 50$ and $n \equiv 2 \ \dot{mod}\ 12$, and in Table 2 for $n \geq 68$ and $n \equiv 8 \ \dot{mod}\ 12$. The smallest size for which these tables are relevant is $n = 50$, which is too large to represent. However, while it is not actually a solution to the (2, 26)-Leprechauns Problem, Fig. 3 provides a convenient visual representation in a manageable size of the construction when placing the first leprechaun in tile (1,8).

From Lemma 3 we know that Lines 5–17 produce an arrangement of n non-attacking leprechauns on an $(n+1) \times n$ board. It remains to demonstrate that the operations done in Lines 21 and 22 are equivalent to moving the leprechaun in the last row to an empty row, and that this move does not introduce a conflict.

From Table 1 (respectively 2), we know that the leprechaun in Column $\frac{2(n+1)}{3} - 3$ (respectively $\frac{2(n+1)}{3} - 5$) was initially placed in Row $12 + 3(\frac{2(n+1)}{3} - 3 - \frac{n+1}{3} - 1) = 12 + 3(\frac{n+1}{3} - 4) = n + 1$ (respectively $18 + 3(\frac{2(n+1)}{3} - 5 - \frac{n+1}{3} - 1) = 18 + 3(\frac{n+1}{3} - 6) = n + 1$). We also know that no leprechaun was placed in Row 13 (respectively 19). Therefore Line 21 (respectively 22) describes moving the leprechaun in Row $n+1$ to an empty row, resulting in an $n \times n$ board with n leprechauns, and all row and column constraints satisfied. Moreover, the tables inform us that the leprechauns in the previous two columns are in Rows $n-5$ and $n-2$, and that the leprechauns in the next two columns are in Rows 3 and 6. Since $n \geq 50$, this means that moving the leprechaun in tile $(\frac{2(n+1)}{3} - 3, n+1)$ (respectively $(\frac{2(n+1)}{3} - 5, n+1)$) to tile $(\frac{2(n+1)}{3} - 3, 13)$ (respectively $(\frac{2(n+1)}{3} - 5, 19)$) does not put it within reach of another range-2 leprechaun. Therefore it only remains to show that no other leprechaun shares a diagonal with the tile $(\frac{2(n+1)}{3} - 3, 13)$ (respectively $(\frac{2(n+1)}{3} - 5, 19)$).

Let $n \geq 50$ such that $n \equiv 2 \ \dot{mod}\ 12$. The leprechaun l that was moved to the tile $(\frac{2(n+1)}{3} - 3, 13)$ is in the sum diagonal $d_+ = \frac{2(n+1)}{3} + 10$ and in the difference diagonal $d_- = \frac{2(n+1)}{3} - 16$. Since $n+1$ is divisible by 3, both d_+ and d_- are even. Let l' be a leprechaun such that l and l' attack each other. Let c' be the column of l', let d'_+ be the sum diagonal of l', and let d'_- be the difference diagonal of l'. The possible values for d'_+ and d'_- depending on c' are given in Table 1.

- If $1 \leq c' \leq \frac{n+1}{3} - 2$: both d'_+ and d'_- are odd, so there can be no conflict.
- If $\frac{n+1}{3} - 1 \leq c' \leq \frac{n+1}{3}$: since $n \geq 50$, we have $d'_+ \leq \frac{n+1}{3} + 5 < \frac{2(n+1)}{3} + 10 = d_+$, so $d'_+ \neq d_+$. Similarly, since $n \geq 50$ we have $d'_- \leq \frac{n+1}{3} - 3 < \frac{2(n+1)}{3} - 16 = d_-$, so $d'_- \neq d_-$.
- If $\frac{n+1}{3} + 1 \leq c' \leq \frac{2(n+1)}{3} - 3$: in this case $d'_+ = 4c' + 8 - n$ and $d'_- = -2c' - 8 + n$. Since $n \equiv 2 \ \dot{mod}\ 12$, and in particular since $n \equiv 2 \ \dot{mod}\ 4$, we have $d'_+ \equiv 2 \ \dot{mod}\ 4$, but $d_+ = \frac{2(n+1)}{3} + 10 \equiv 4 \ \dot{mod}\ 4$, so $d'_+ \neq d_+$. Since $c' \geq \frac{n+1}{3} + 1$, we have $d'_- \leq -2(\frac{n+1}{3} + 1) - 8 + n = \frac{n+1}{3} - 11$, which is strictly smaller than $\frac{2(n+1)}{3} - 16 = d_-$ because $n \geq 50$, so $d'_- \neq d_-$.
- If $\frac{2(n+1)}{3} - 2 \leq c' \leq \frac{2(n+1)}{3}$: both d'_+ and d'_- are odd, so there can be no conflict.
- If $\frac{2(n+1)}{3} + 1 \leq c' \leq n - 4$: both d'_+ and d'_- are odd, so there can be no conflict.

- If $n-3 \leq c' \leq n$: since $n \geq 50$, we have $d_+ = \frac{2(n+1)}{3} + 10 < n-2 \leq d'_+$, so $d'_+ \neq d_+$. Similarly, since $n \geq 50$ we have $d_- = \frac{2(n+1)}{3} - 16 < n - 10 \leq d'_-$, so $d'_- \neq d_-$.

Therefore, if $n \geq 50$ and $n \equiv 2 \ m\dot{o}d\ 12$, then the leprechaun l does not share a diagonal with another leprechaun.

If $n \geq 68$ and $n \equiv 8 \ m\dot{o}d\ 12$, the proof is almost identical with the minor difference being that if $\frac{n+1}{3} + 1 \leq c' \leq \frac{2(n+1)}{3} - 5$, the congruences mȯdulo 4 of d'_+ and d_+ are switched. In any case, the result is the same: moving the leprechaun in Row $n+1$ to Row 19 in Line 22 does not violate any diagonal constraint. Since that was all that remained to show, this concludes the proof. □

Now that we have treated each residue class mȯdulo 6, we collect them together in Algorithm 5, which solves the $(2, n)$-Leprechauns Problem for all n.

Data: An integer n.
Result: Either a solution to the $(2, n)$-Leprechauns Problem or "No solution".
1 **if** $n \geq 2$ *and* $n \leq 9$ **then return** "No solution";
2 **if** $n = 14$ **then return** (1,4,7,10,13,5,8,11,14,2,6,9,3,12);
3 **if** $n \equiv 1 \ m\dot{o}d\ 6$ **or** $n \equiv 5 \ m\dot{o}d\ 6$ **then return** Algorithm 1;
4 **if** $n \equiv 4 \ m\dot{o}d\ 6$ **or** $n \equiv 6 \ m\dot{o}d\ 6$ **then return** Algorithm 2;
5 **if** $n \equiv 3 \ m\dot{o}d\ 6$ **then return** Algorithm 3;
6 **if** $n \equiv 2 \ m\dot{o}d\ 6$ **then return** Algorithm 4;

Algorithm 5: General case.

Theorem 1. *Algorithm 5 solves the $(2, n)$-Leprechauns Problem in linear time.*

Proof. Each of Algorithms 1, 2, 3 and 4 loops through Columns 1 to n exactly once, so Algorithm 5's time complexity is clearly linear. As for its correctness:

Line 1: Already known result [10].
Line 2: Fig. 1 shows that this is indeed a solution.
Line 3: From Lemma 1.
Line 4: From Lemma 2.
Line 5: From Lemma 3.
Line 6: From Lemma 4. □

4. The ($k \geq 3$, n)-Leprechauns problem

4.1. When n is under $(k+1)^2$

We know that the $(1, n)$-Leprechauns (n-Queens) Problem has a solution for $n = 1$, has no solution for $n = 2$ to $n = 3$, and has a solution for $n \geq 4$. We also now know that the $(2, n)$-Leprechauns Problem has a solution for $n = 1$, has no solution for $n = 2$ to $n = 9$, and has a solution for $n \geq 10$. This would seem to indicate a pattern in the behavior of the (k, n)-Leprechauns Problem, where for $n = 1$ there is a trivial solution, for n from 2 to some N the board is too small to support a solution, and finally for sizes over N the board is large enough to accommodate leprechaun distance constraints. We suspect that the transition between these two phases occurs around $n = (k+1)^2$. To support this conjecture, we are now going to present two formal results that demonstrate the importance of this particular function. First we show that the whole area under this parabola is in the unsatisfiability region (apart from the trivial case $n = 1$).

Proposition 1. *Let $k > 0$ and $n > 1$ be two integers such that $n < (k+1)^2$. Then there is no solution for the (k, n)-Leprechauns Problem.*

Proof. Let $n' = \lceil \frac{n}{k+1} \rceil$ and for $1 \leq i, j \leq n'$, let the box $B_{i,j}$ be the set of tiles with coordinates (i', j') such that $(k+1) \times (i-1) + 1 \leq i' \leq \min((k+1) \times (i-1) + k + 1, n)$ and $(k+1) * (j-1) + 1 \leq j' \leq \min((k+1) \times (j-1) + k + 1, n)$. The dimensions of a box $B_{i,j}$ are at most $k+1 \times k+1$, so any two tiles in $B_{i,j}$ are at distance at most k from each other, so at most one range-k leprechaun can be placed in a box $B_{i,j}$.

Consider the set of boxes $\mathcal{B}$ that contains the boxes $B_{i,j}$. If $k+1$ divides n then all n'^2 boxes in $\mathcal{B}$ will be of dimension $(k+1) \times (k+1)$, otherwise $\mathcal{B}$ will contain $(n'-1)^2$ boxes of dimension $(k+1) \times (k+1)$, $n'-1$ boxes of dimension $k' \times (k+1)$, $n'-1$ boxes of dimension $(k+1) \times k'$ and one box of dimension $k' \times k'$ for some $1 \leq k' < k+1$. Moreover $\mathcal{B}$ forms a partition of the board, illustrated in Fig. 4.

If $n' < k+1$, then the number of boxes is fewer than the number of leprechauns to place, so we have $n' = k+1$.

Since at most one range-k leprechaun can fit in each box, and since the $k+1$ boxes $B_{1,1}, B_{1,2}, \ldots, B_{1,n'}$ span exactly $k+1$ columns of the board, we know that each of these boxes contains exactly one leprechaun. In particular the box

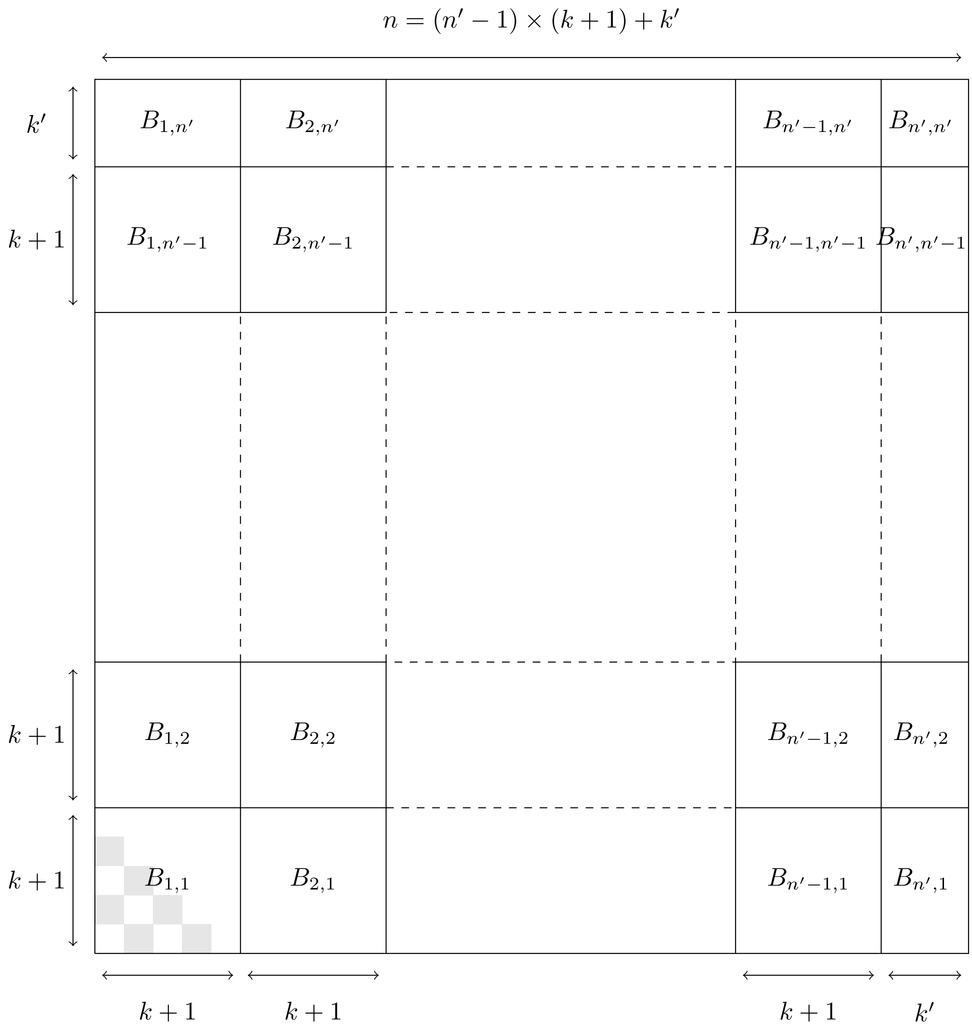

Fig. 4. Partition of the board into boxes.

$B_{1,n'}$ contains one leprechaun. With the same reasoning on the other vertical sets of boxes, we know that the other boxes $B_{2,n'}, \ldots, B_{n'-1,n'}$ at the top edge of the board (minus the last one which spans only k' columns) contain one range-k leprechaun each. So these k boxes ($n' = k + 1$ boxes at the top edge of the board, minus the last one) contain k total range-k leprechauns. But they span k' rows, so they can contain at most k' total range-k leprechauns. So $k' \geq k$. Since $n < (k+1)^2$ we have $k' < k + 1$ and therefore $k' = k$.

Since $k' = k$, we have $n = k(k+1) + k = (k+1)^2 - 1$, with the number of boxes in $\mathcal{B}$ being $(k+1)^2$. So every box in $\mathcal{B}$ but one contains a range-k leprechaun. Since the $k+1$ boxes $B_{1,n'}, B_{2,n'}, \ldots, B_{n',n'}$ at the top edge of the board only span $k' = k$ rows of the board, it means that the empty box is among them. Similarly, since the $k+1$ boxes $B_{n',1}, B_{n',2}, \ldots, B_{n',n'}$ at the right edge of the board only span k columns of the board, it means that the empty box is also among them. So the empty box is $B_{n',n'}$ and every other box in $\mathcal{B}$ contains exactly one range-k leprechaun.

Since the box $B_{1,n'}$ spans only k rows of the board, the range-k leprechaun in that particular box attacks all tiles on the top row of the box $B_{1,n'-1}$, which is the intersection of $B_{1,n'-1}$ with the $n - k'$th row of the board. So there is no range-k leprechaun on the top row of the box $B_{1,n'-1}$. Similarly, there is no range-k leprechaun on the top row of the boxes $B_{2,n'-1}, \ldots, B_{n'-1,n'-1}$. So the range-$k$ leprechaun l in the $n - k'$th row of the board is in the top row of the box $B_{n',n'-1}$. Using a mirrored reasoning we also get that the range-k leprechaun l' in the $n - k'$th column of the board is in the rightmost column of the box $B_{n'-1,n'}$. Since l is at most k columns to the right of l', and since l' is at most k rows above l, the two range-k leprechauns attack each other. So there is no solution for the (k, n)-Leprechauns Problem when $n < (k+1)^2$. □

Note that at no point in the proof of Proposition 1 did we mention diagonal constraints. These only come in play for $n \geq (k+1)^2$, which is where the board is large enough to contain an arrangement that fulfills all other constraints. As we

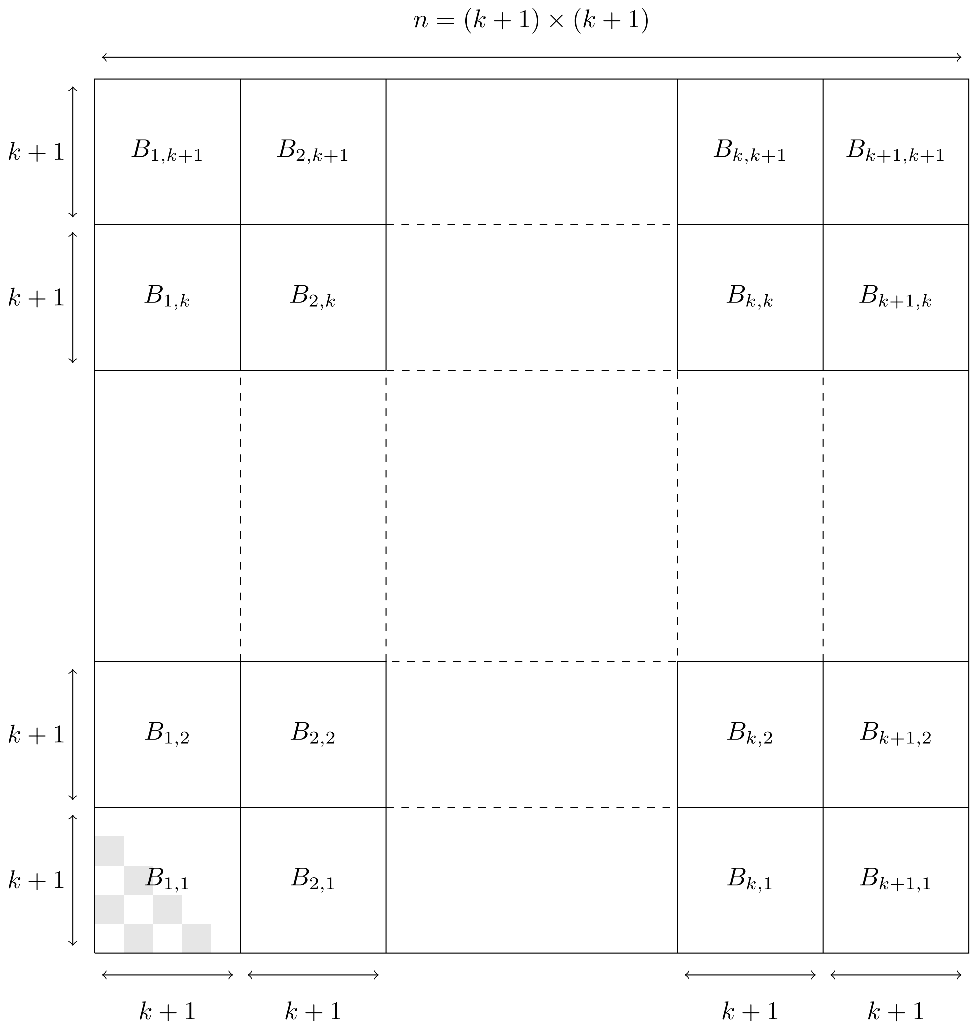

Fig. 5. Partition of the board into boxes when $n = (k+1)^2$.

are now going to show, $(k+1)^2$ is the first size for which the (k, n)-Leprechauns Problem is neither always satisfiable nor always unsatisfiable. Instead, whether there is a solution depends on the parity of k. We will also characterize solutions, when they exist.

Proposition 2. *Let $k > 0$ and n be two integers such that $n = (k+1)^2$. Then there is a solution to the (k, n)-Leprechauns Problem if and only if k is odd, in which case there are exactly two (symmetrical) solutions.*

Proof. We define boxes the same way we did in the proof for Proposition 1: each box $B_{i,j}$ is a square subset of the board containing $(k+1) \times (k+1)$ tiles, and the bottom left tile of $B_{i,j}$ is the tile $((k+1) \times (i-1) + 1, (k+1) \times (j-1) + 1)$. Since $n = (k+1)^2$, there are n boxes, as illustrated in Fig. 5. Also, since each box can contain at most one leprechaun and since n leprechauns need to be placed on the board, each box contains exactly one leprechaun.

We start by showing that within a column of boxes, all leprechauns must be located at the same row of their respective boxes.

Lemma 5. *Let i be such that $1 \leq i \leq k+1$. Then there exist some r such that for each $1 \leq j \leq k+1$, there is a leprechaun in the rth row of $B_{i,j}$.*

Proof. We know that there is a leprechaun on Row $k+1$. Let i_A such that the box $B_{i_A,1}$ is the bottom box that contains a leprechaun in its $k+1$th row. Since leprechauns have a range of k, there cannot be a leprechaun in the first k rows of the box $B_{i_A,2}$. So $B_{i_A,2}$ also contains a leprechaun in its $k+1$th row. Using the same argument for boxes $B_{i_A,3}$ to $B_{i_A,k+1}$, we can see that all boxes $B_{i_A,j}$ contain a leprechaun in their $k+1$th row.

We know that there is a leprechaun on Row k. Let i_B such that $i_B \neq i_A$ and the box $B_{i_B,1}$ is the bottom box that contains a leprechaun in its kth row. Since leprechauns have a range of k, there cannot be a leprechaun in the first $k-1$ rows of the box $B_{i_B,2}$. So there is a leprechaun in either the kth or the $k+1$th row of $B_{i_B,2}$. But we know that $B_{i_A,2}$ is the box (among the ones on the second row of boxes) that contains a leprechaun in its $k+1$th row. Therefore $B_{i_B,2}$ contains a leprechaun in its kth row. Using the same argument for boxes $B_{i_B,3}$ to $B_{i_B,k+1}$, we can see that all boxes $B_{i_B,j}$ contain a leprechaun in their kth row.

Repeating the reasoning for Rows $k-1$ to 1 completes the proof of the Lemma. □

By switching row and column coordinates the same proof can be used to prove the corresponding property for leprechauns within the same row of boxes:

Lemma 6. *Let j be such that $1 \leq j \leq k+1$. Then there exist some c such that for each $1 \leq i \leq k+1$, there is a leprechaun in the cth column of $B_{i,j}$.*

We now show that within a column of boxes, the columns of the leprechauns either monotonically increase or monotonically decrease.

Lemma 7. *Let i be such that $1 \leq i \leq k+1$. Then one of the following is true:*

- *For all $1 \leq j \leq k+1$, there is a leprechaun in the jth column of $B_{i,j}$.*
- *For all $1 \leq j \leq k+1$, there is a leprechaun in the $k+2-j$th column of $B_{i,j}$.*

Proof. From Lemma 5 we know that there exists some r such that all leprechauns in this column of boxes are in the rth row of their columns.

Suppose that the order of the leprechauns' columns is not monotonically decreasing. Therefore there exist j_0, c_1 and c_2 such that $c_1 < c_2$, there is a leprechaun in the c_1th column of B_{i,j_0}, and there is a leprechaun in the c_2th column of B_{i,j_0+1}. Suppose also that the order of the leprechauns' rows in the row of boxes j_0 is not monotonically decreasing. Then there exist i_0, r_1 and r_2 such that $r_1 < r_2$, there is a leprechaun in the r_1th row of B_{i_0,j_0}, and there is a leprechaun in the r_2th row of B_{i_0+1,j_0}. Since there is a leprechaun in the c_1th column of B_{i,j_0}, we know from Lemma 6 that the leprechaun in the r_2th row of B_{i_0+1,j_0} is in the c_1th column of this box. Since there is a leprechaun in the c_2th column of B_{i,j_0+1} and there is a leprechaun in the r_1th row of B_{i_0,j_0}, we also know from Lemmas 5 and 6 that there is a leprechaun in the r_1th row and c_2th column of B_{i_0,j_0+1}. Since $c_1 < c_2$ and $r_1 < r_2$, this leprechaun is attacking the leprechaun located in the r_2th row and c_1th column of B_{i_0+1,j_0}. So if the order of the leprechauns' columns is not monotonically decreasing, then the order of the leprechauns' rows must be monotonically decreasing.

By replacing "decreasing" with "increasing" and "$<$" with "$>$" in the previous paragraph, we can also show that if the order of the leprechauns' columns is not monotonically increasing, then the order of the leprechauns' rows must be monotonically increasing. So if the order of the leprechauns' columns is neither monotonically decreasing nor monotonically increasing then we have a contradiction. So the order of the leprechauns' columns is either monotonically decreasing or monotonically increasing. □

As before, switching row and columns gives us a proof for the corresponding property on rows of boxes:

Lemma 8. *Let j be such that $1 \leq j \leq k+1$. Then one of the following is true:*

- *For all $1 \leq i \leq k+1$, there is a leprechaun in the ith row of $B_{i,j}$.*
- *For all $1 \leq i \leq k+1$, there is a leprechaun in the $k+2-i$th row of $B_{i,j}$.*

At this point, we have four remaining potential solutions:

1. The order of the leprechauns' columns is monotonically increasing and the order of the leprechauns' row is monotonically decreasing: for all $1 \leq i, j \leq k+1$, there is a leprechaun at coordinates $(j, k+2-i)$ in the box $B_{i,j}$, corresponding to the board tile $((k+1)\times(i-1)+j, (k+1)\times(j-1)+k+2-i)$.
2. The order of the leprechauns' columns is monotonically decreasing and the order of the leprechauns' rows is monotonically increasing: for all $1 \leq i, j \leq k+1$, there is a leprechaun at coordinates $(k+2-j, i)$ in the box $B_{i,j}$, corresponding to the board tile $((k+1)\times(i-1)+k+2-j, (k+1)\times(j-1)+i)$.
3. The order of the leprechauns' columns is monotonically increasing and the order of the leprechauns' rows is monotonically increasing: for all $1 \leq i, j \leq k+1$, there is a leprechaun at coordinates (j, i) in the box $B_{i,j}$, corresponding to the board tile $((k+1)\times(i-1)+j, (k+1)\times(j-1)+i)$.
4. The order of the leprechauns' columns is monotonically decreasing and the order of the leprechauns' rows is monotonically decreasing: for all $1 \leq i, j \leq k+1$, there is a leprechaun at coordinates $(k+2-j, k+2-i)$ in the box $B_{i,j}$, corresponding to the board tile $((k+1)\times(i-1)+k+2-j, (k+1)\times(j-1)+k+2-i)$.

In the third potential solution, the leprechaun in the box $B_{2,1}$ is on the board tile $(k+2, 2)$ and the leprechaun in the box $B_{1,2}$ is on the board tile $(2, k+2)$. These two leprechauns are attacking each other, so the third potential solution is

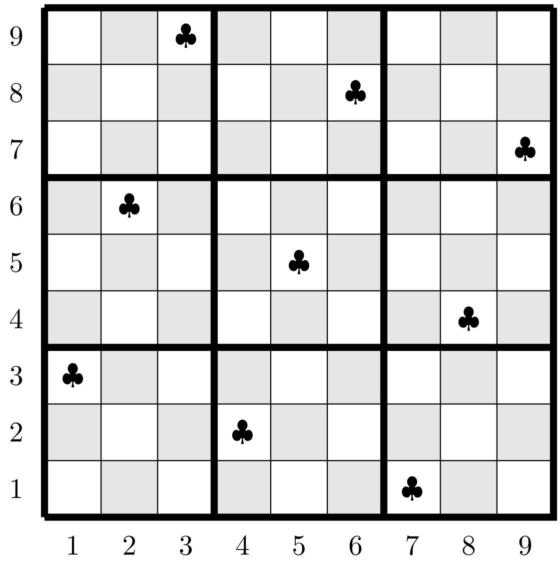

Fig. 6. Potential solution for $k = 2$ and $n = (k + 1)^2 = 9$.

not a solution. In the fourth potential solution, the leprechaun in the box $B_{1,1}$ is on the board tile $(k + 1, k + 1)$ and the leprechaun in the box $B_{2,2}$ is on the board tile $(2k + 1, 2k + 1)$. These two leprechauns are attacking each other, so the fourth potential solution is not a solution.

Now only two potential solutions remain. In the first one, the leprechaun in the box $B_{1,1}$ is on the board tile $(1, k+1)$. In the second one, the leprechaun in the box $B_{1,1}$ is on the board tile $(k + 1, 1)$. Therefore the two potential solutions are distinct. We now show that they mirror each other through the main diagonal.

Lemma 9. *Let i_0 and j_0 be two integers such that $1 \leq i_0, j_0 \leq n$. Then there is a leprechaun on tile (i_0, j_0) in the first potential solution if and only if there is a leprechaun on tile (j_0, i_0) in the second potential solution.*

Proof. $\Rightarrow$ Suppose that there is a leprechaun on tile (i_0, j_0) in the first potential solution. Then there are $1 \leq i, j \leq k + 1$ such that $i_0 = (k+1)\times(i-1)+j$ and $j_0 = (k+1)\times(j-1)+k+2-i$. Let i' and j' be such that $i' = j$ and $j' = i$. We know that there is a leprechaun on tile $((k + 1) \times (i' - 1) + k + 2 - j', (k + 1) \times (j' - 1) + i')$ in the second potential solution. Since $i' = j$ and $j' = i$, we have $(k + 1) \times (i' - 1) + k + 2 - j' = j_0$ and $(k + 1) \times (j' - 1) + i' = i_0$. So there is a leprechaun on tile (j_0, i_0) in the second potential solution.

$\Leftarrow$ Since each potential solution has the same number of leprechauns, we only needed to prove one direction. $\square$

Since both remaining potential solutions are symmetrical, we will from now on only consider the first one. For ease of future reference, we give in Table 3 the coordinates in this potential solution of the leprechaun in $B_{i,j}$ and of its nearest neighbors. We also illustrate the potential solution for an even value of k in Fig. 6 and for an odd value in Fig. 7. In the latter case, no two leprechauns are attacking each other, and we therefore have an actual solution for $k = 3$ and $n = (k + 1)^2 = 16$. In the former case however, there are several cases of leprechauns attacking each other, for example the two leprechauns in $(1, 3)$ and $(6, 8)$. This proves that there is no solution to the $(2, 9)$-Leprechauns Problem. More generally:

Lemma 10. *Let $k > 0$ and n be two integers such that k is even and $n = (k + 1)^2$. Then in the first potential solution to the (k, n)-Leprechauns Problem, the leprechauns in boxes $B_{1,k/2}$ and $B_{k/2+1,k+1}$ attack each other.*

Proof. Let l_1 be the leprechaun in the box $B_{1,k/2}$ and let l_2 be the leprechaun in the box $B_{k/2+1,k+1}$. We know from Table 3 that the coordinates of l_1 are (i_1, j_1) with $i_1 = (k + 1) \times (1 - 1) + k/2$ and $j_1 = (k + 1) \times (k/2 - 1) + k + 2 - 1$, while the coordinates of l_2 are (i_2, j_2) with $i_2 = (k + 1) \times (k/2 + 1 - 1) + k + 1$ and $j_2 = (k + 1) \times (k + 1 - 1) + k + 2 - (k/2 + 1)$. Therefore

$$\begin{aligned} j_1 - i_1 &= (k+1)\times(k/2-1)+k+2-1-(k+1)\times(1-1)-k/2 \\ &= (k + 1) \times k/2 - (k + 1) + k + 1 - k/2 \\ &= (k + 1) \times k/2 - k/2 \\ &= k \times k/2 \end{aligned}$$

Table 3
Coordinates in the first potential solution of the leprechaun l in the box $B_{i,j}$ and of the leprechauns in the surrounding boxes.

Box	Coordinates	Difference with l's coordinates	Distance to l
$B_{i,j}$	$((k+1)\times(i-1)+j,(k+1)\times(j-1)+k+2-i)$	$(0,0)$	0
$B_{i+1,j}$	$((k+1)\times i+j,(k+1)\times(j-1)+k+1-i)$	$(k+1,-1)$	k+1
$B_{i+1,j+1}$	$((k+1)\times i+j+1,(k+1)\times j+k+1-i)$	$(k+2,k)$	k+2
$B_{i,j+1}$	$((k+1)\times(i-1)+j+1,(k+1)\times j+k+2-i)$	$(1,k+1)$	k+1
$B_{i-1,j+1}$	$((k+1)\times(i-2)+j+1,(k+1)\times j+k+3-i)$	$(-k,k+2)$	k+2
$B_{i-1,j}$	$((k+1)\times(i-2)+j,(k+1)\times(j-1)+k+3-i)$	$(-k-1,1)$	k+1
$B_{i-1,j-1}$	$((k+1)\times(i-2)+j-1,(k+1)\times(j-2)+k+3-i)$	$(-k-2,-k)$	k+2
$B_{i,j-1}$	$((k+1)\times(i-1)+j-1,(k+1)\times(j-2)+k+2-i)$	$(-1,-k-1)$	k+1
$B_{i+1,j-1}$	$((k+1)\times i+j-1,(k+1)\times(j-2)+k+1-i)$	$(k,-k-2)$	k+2

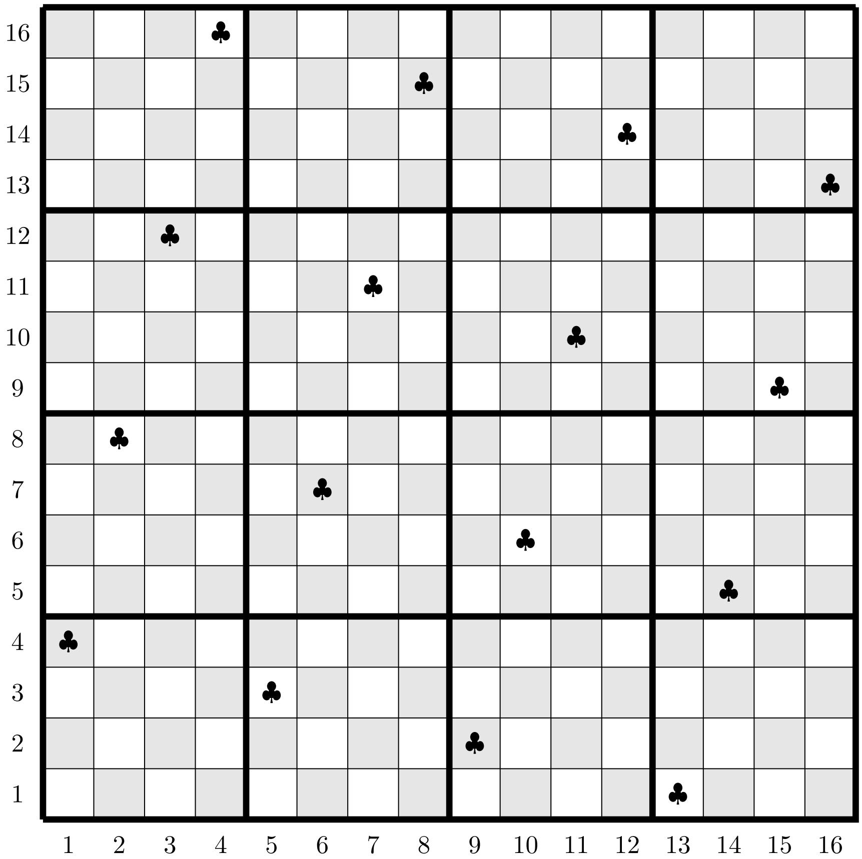

Fig. 7. Potential solution for $k=3$ and $n=(k+1)^2=16$.

and

$$\begin{aligned} j_2-i_2 &= (k+1)\times(k+1-1)+k+2-(k/2+1)-(k+1)\times(k/2+1-1)-k-1 \\ &= (k+1)\times k+k+2-k/2-1-(k+1)\times(k/2)-k-1 \\ &= (k+1)\times k-k/2-(k+1)\times(k/2) \\ &= (k+1)\times(k/2)-k/2 \\ &= k\times k/2 \end{aligned}$$

So $j_1-i_1=j_2-i_2$. So l_1 and l_2 are on the same diagonal. So l_1 and l_2 are attacking each other. □

So if k is even, then the only potential solution (modulo reflection) to the (k,n)-Leprechauns Problem is not an actual solution. This completes the proof of the Proposition for the case when k is even. Consequently we now assume that k is odd.

One of the consequences of Lemma 8 is that there is a leprechaun in each row of the board, so the row constraints of the problem are satisfied. Similarly, from Lemma 7 we know that there is a leprechaun in each column of the board and therefore that the column constraints of the problem are satisfied.

As can be seen in Table 3, for all $1 \leq i, j \leq k+1$, the range-k leprechaun in the box $B_{i,j}$ is out of reach of the range-k leprechauns in the surrounding boxes. Therefore the proximity constraints are also satisfied.

We are now going to show that the last remaining constraints, the diagonal ones, are satisfied by the first potential solution. Let l be the leprechaun in the box $B_{i,j}$ with $1 \leq i, j \leq k+1$ and let l' be the leprechaun in the box $B_{i',j'}$ with $1 \leq i', j' \leq k+1$ such that either $i' \neq i$ or $j' \neq j$. Let d_+ (respectively d_-) be the sum (respectively difference) of l's coordinates, and let d'_+ (respectively d'_-) be the sum (respectively difference) of l''s coordinates.

Suppose that $d'_+ = d_+$. From Table 3 we have:

$$(k+1) \times (i'-1) + j' + (k+1) \times (j'-1) + k + 2 - i' = (k+1) \times (i-1) + j + (k+1) \times (j-1) + k + 2 - i$$
$$\Downarrow$$
$$(k+1) \times i' + j' + (k+1) \times j' - i' = (k+1) \times i + j + (k+1) \times j - i$$
$$\Downarrow$$
$$k \times i' + (k+2) \times j' = k \times i + (k+2) \times j$$
$$\Downarrow$$
$$k \times (i'+j') + 2j' = k \times (i+j) + 2j$$

So $(2j' \ \dot{mod}\ k) = (2j \ \dot{mod}\ k)$. So either $(j' \ \dot{mod}\ k) = (j \ \dot{mod}\ k)$ or $(j' \ \dot{mod}\ k) = ((j + k/2) \ \dot{mod}\ k)$. Since k is odd, we have $(j' \ \dot{mod}\ k) = (j \ \dot{mod}\ k)$. Moreover $j' \neq j$ (because otherwise we would also have $i' = i$ and we assumed either $i' \neq i$ or $j' \neq j$). We have $1 \leq j, j' \leq k+1$, so either $j = 1$ and $j' = k+1$ or the other way around. Without loss of generality, assume the former. The equality above becomes then:

$$k \times (i' + k + 1) + 2(k+1) = k \times (i+1) + 2$$
$$\Downarrow$$
$$k \times (i' + k) + 2k = k \times i$$

Dividing by k we get $i = i' + k + 2$. Since $1 \leq i, i' \leq k+1$, we have a contradiction. Therefore d'_+ cannot be equal to d_+.

Suppose now that $d'_- = d_-$. From Table 3 we have:

$$(k+1) \times (i'-1) + j' - (k+1) \times (j'-1) - k - 2 + i' = (k+1) \times (i-1) + j - (k+1) \times (j-1) - k - 2 + i$$
$$\Downarrow$$
$$(k+1) \times i' + j' - (k+1) \times j' + i' = (k+1) \times i + j - (k+1) \times j + i$$
$$\Downarrow$$
$$(k+2) \times i' - k \times j' = (k+2) \times i - k \times j$$
$$\Downarrow$$
$$k \times (i'-j') + 2i' = k \times (i-j) + 2i$$

So $(2i' \ \dot{mod}\ k) = (2i \ \dot{mod}\ k)$. So either $(i' \ \dot{mod}\ k) = (i \ \dot{mod}\ k)$ or $(i' \ \dot{mod}\ k) = ((i + k/2) \ \dot{mod}\ k)$. Since k is odd, we have $(i' \ \dot{mod}\ k) = (i \ \dot{mod}\ k)$. Moreover $i' \neq i$ (because otherwise we would also have $j' = j$ and we assumed either $i' \neq i$ or $j' \neq j$). We have $1 \leq i, i' \leq k+1$, so either $i = 1$ and $i' = k+1$ or the other way around. Without loss of generality, assume the former. The equality above becomes then:

$$k \times (k + 1 - j') + 2(k+1) = k \times (1-j) + 2$$
$$\Downarrow$$
$$k \times (k - j') + 2k = k \times (-j)$$

Dividing by k we get $k - j' + 2 = -j$, or equivalently $j = j' - k - 2$. Since $1 \leq j, j' \leq k+1$, we have a contradiction. Therefore d'_- cannot be equal to d_-. Therefore the diagonal constraints are satisfied, and the first potential solution is an actual solution to the (k, n)-Leprechauns Problem when k is odd and $n = (k+1)^2$. Since the second potential solution is a reflective image of the first potential solution, it is an actual solution as well. This completes the proof of the Proposition. □

We can combine the last two propositions to completely solve the (k, n)-Leprechauns Problem for all configurations of k and n such that n is at or below the parabola corresponding to the function $(k+1)^2$.

Theorem 2. *Let $k > 0$ and $n > 0$ be such that $n \leq (k+1)^2$. Then the (k, n)-Leprechauns Problem is satisfiable if and only if k is odd and $n = (k+1)^2$, in which case there are exactly two symmetrical solutions, or if $n = 1$, in which case there is exactly one trivial solution.*

Table 4
Smallest non-trivial board for which there is a solution to the (k, n)-Leprechauns Problem.

k	$(k+1)^2$	n	offset
1	4	4	0
2	9	10	1
3	16	16	0
4	25	28	3
5	36	36	0
6	49	52	3
7	64	64	0
8	81	82	1
9	100	100	0

Table 5
Number p of solutions to the (k, n)-Leprechauns Problem for some combinations of k and n. For each k, the first value of n shown is the first $n > 1$ for which there exists at least one solution.

$k = 3$		$k = 4$		$k = 5$		$k = 6$	
n	p	n	p	n	p	n	p
16	2	28	10	36	2	52	6
17	34	29	286	37	74		
18	4	30	696	38	0		
19	112	31	10016	39	48		
20	516	32	201332	40	68		
21	7312			41	808		
22	81324						
23	1056560						
24	13443944						
25	171919446						

We have fully treated the case when $n \leq (k+1)^2$, and proven that it covers the board sizes where the additional distance constraints introduced by leprechauns cannot be satisfied.

4.2. When n is over $(k+1)^2$

It is natural to wonder what is for each k the first non-trivial (that is, not 1) n for which there exists a solution to the (k, n)-Leprechauns Problem. From Theorem 2, we know that for odd k the answer is exactly $(k+1)^2$. Since a solution to the $(k+1, n)$-Leprechauns Problem also satisfies all constraints from the (k, n)-Leprechauns Problem, this implies that the demarcation lies between $(k+1)^2$ and $(k+2)^2$ for all k.

For the first few values of k, we collect in Table 4 the smallest size $n > 1$ for which a board is large enough to admit a solution to the (k, n)-Leprechauns Problem. The entries for odd k were filled by using the result from Theorem 2. For even k, at least up until 8, the smallest value of n (> 1) for which the (k, n)-Leprechauns Problem is satisfiable appears to be equal to the smallest prime number larger than $(k+1)^2$, minus 1. That this value is an upper bound comes from a construction for the n-Queens Problem when n is either prime or one less than a prime [24]. We can, in a manner similar to the proofs of Lemmas 1 and 3, generalize this construction to the (k, n)-Leprechauns Problem when n is big enough. To verify that this is indeed the smallest eligible n, we used exhaustive search to look for a solution to the (k, n)-Leprechauns for each n between the known bounds, and did not find any.

Interestingly, while for $k = 1$ and $k = 2$ there is only one continuous unsatisfiability phase (from 2 to 3 and from 2 to 9 respectively), this is not true for all k. For the $(5, n)$-Leprechauns Problem in particular, there is no solution for $2 \leq n \leq 35$, there is a solution for $n = 36$ and for $n = 37$, but there is no solution for $n = 38$.

We give in Table 5 the number of solutions to the (k, n)-Leprechauns Problem for the first few values of $k \geq 3$ and $n \geq (k+1)^2$. These numbers were provided by an anonymous reviewer, to whom we are grateful. Similar results exist for the (1,n)- (queens [26]) and (2,n)- (superqueens [25]) Leprechauns Problems.

5. Conclusion

We have made the first strides towards solving the Diverse n-Queens Problem by studying the equivalent (k, n)-Leprechauns Problems. We have given a vertical result, in the form of an algorithm that can solve the $(2, n)$-Leprechauns Problem, as well as a horizontal one, in our characterization of the Problem for $n \leq (k+1)^2$. The latter provides strong theoretical evidence that the phase transition from unsatisfiability to satisfiability in the (k, n)-Leprechauns Problem occurs around $n = (k+1)^2$.

In addition of diversity, potential future work on the (k, n)-Leprechauns Problem could try to apply it on representation, which is a related but different notion that has been studied in other constraint fields [23].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This material is based upon works supported by the Science Foundation Ireland under Grant No. 12/RC/2289-P2 which is co-funded under the European Regional Development Fund.

References

[1] Ola Angelsmark, Johan Thapper, Algorithms for the maximum Hamming distance problem, in: Boi Faltings, Adrian Petcu, François Fages, Francesca Rossi (Eds.), Recent Advances in Constraints, Joint ERCIM/CoLogNet International Workshop on Constraint Solving and Constraint Logic Programming, CSCLP 2004, Lausanne, Switzerland, June 23–25, 2004, in: Lecture Notes in Computer Science, vol. 3419, Springer, 2004, pp. 128–141, Revised Selected and Invited Papers.
[2] Jordan Bell, Brett Stevens, A survey of known results and research areas for n-queens, Discrete Math. 309 (1) (2009) 1–31.
[3] Hans Bodlaender, Fergus Duniho, The chess variant pieces, 2020, https://www.chessvariants.com/piececlopedia.dir/amazon.html.
[4] Pierluigi Crescenzi, Gianluca Rossi, On the Hamming distance of constraint satisfaction problems, Theoret. Comput. Sci. 288 (1) (2002) 85–100.
[5] Vilhelm Dahllöf, Algorithms for max Hamming exact satisfiability, in: Xiaotie Deng, Ding-Zhu Du (Eds.), Algorithms and Computation, 16th International Symposium, ISAAC 2005, Sanya, Hainan, China, December 19–21, 2005, Proceedings, in: Lecture Notes in Computer Science, vol. 3827, Springer, 2005, pp. 829–838.
[6] Cengiz Erbas, Murat M. Tanik, Zekeriya Aliyazicioglu, Linear congruence equations for the solutions of the N-queens problem, Inf. Process. Lett. 41 (6) (1992) 301–306.
[7] Edward Falkener, Games Ancient and Oriental, and how to Play Them: Being the Games of the Greek, the Ludus Latrunculorum of the Romans and the Oriental Games of Chess, Draughts, Backgammon and Magic Squares, Longmans, Green and Company, 1892.
[8] Matteo Fischetti, Domenico Salvagnin, Chasing first queens by integer programming, in: Willem Jan van Hoeve (Ed.), Integration of Constraint Programming, Artificial Intelligence, and Operations Research - 15th International Conference, CPAIOR 2018, Delft, the Netherlands, June 26–29, 2018, Proceedings, in: Lecture Notes in Computer Science, vol. 10848, Springer, 2018, pp. 232–244.
[9] Ian P. Gent, Christopher Jefferson, Peter Nightingale, Complexity of n-queens completion, J. Artificial Intelligence Res. 59 (2017) 815–848.
[10] Evgenii I. Akovlevich Gik, Matematika na shakhmatnoi doske, Nauka, 1976.
[11] Solomon W. Golomb, Sphere packing, coding metrics and chess puzzles, in: Proc. Second Chapel Hill Conf. on Combinatorial Mathematics and its Applications, 1970, pp. 176–189.
[12] Harry Golombeck, Golombeck's Encyclopedia of Chess, Crown Publishers, New York, 1977.
[13] Tarik Hadzic, Alan Holland, Barry O'Sullivan, Reasoning about optimal collections of solutions, in: Ian P. Gent (Ed.), Principles and Practice of Constraint Programming - CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20–24, 2009, Proceedings, in: Lecture Notes in Computer Science, vol. 5732, Springer, 2009, pp. 409–423.
[14] Emmanuel Hebrard, Brahim Hnich, Barry O'Sullivan, Toby Walsh, Finding diverse and similar solutions in constraint programming, in: Manuela M. Veloso, Subbarao Kambhampati (Eds.), Proceedings, the Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9–13, 2005, Pittsburgh, Pennsylvania, USA, AAAI Press / The MIT Press, 2005, pp. 372–377.
[15] Emmanuel Hebrard, Dániel Marx, Barry O'Sullivan, Igor Razgon, Soft constraints of difference and equality, J. Artificial Intelligence Res. 41 (2011) 97–130.
[16] Emmanuel Hebrard, Barry O'Sullivan, Toby Walsh, Distance constraints in constraint satisfaction, in: Manuela M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6–12, 2007, 2007, pp. 106–111.
[17] Pascal Van Hentenryck, Carleton Coffrin, Boris Gutkovich, Constraint-based local search for the automatic generation of architectural tests, in: Ian P. Gent (Ed.), Principles and Practice of Constraint Programming - CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20–24, 2009, Proceedings, in: Lecture Notes in Computer Science, vol. 5732, Springer, 2009, pp. 787–801.
[18] Paul Monsky, E3162, Amer. Math. Monthly 96 (3) (1989) 258–259.
[19] E. Pauls, Das Maximalproblem der Damen auf dem Schachbrete, Dtsch. Schachzeitung 29 (5) (1874) 261–263.
[20] E. Pauls, Das Maximalproblem der Damen auf dem Schachbrete, II, Dtsch. Schachzeitung 29 (9) (1874) 257–267.
[21] Thierry Petit, Andrew C. Trapp, Finding diverse solutions of high quality to constraint optimization problems, in: Qiang Yang, Michael J. Wooldridge (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25–31, 2015, AAAI Press, 2015, pp. 260–267.
[22] George Pólya, Uber die "doppelt-periodischen" Lösungen des n-Damen-problems, W. Ahrens Math. Unterhaltungen Spiele 1 (1921) 364–374.
[23] Nicolas Schwind, Tenda Okimoto, Maxime Clement, Katsumi Inoue, Representative solutions for multi-objective constraint optimization problems, in: Chitta Baral, James P. Delgrande, Frank Wolter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25–29, 2016, AAAI Press, 2016, pp. 601–604.
[24] G. Sforza, Una regola pel gioco della n regine quando n è primo, Period. Mat. Organo Della Math. Soc. Ital. Sci. Math. Fis. 4 (1925) 107–109.
[25] Neil James Alexander Sloane, The on-line encyclopedia of integer sequences, 2020, https://oeis.org/A051223.
[26] Neil James Alexander Sloane, The on-line encyclopedia of integer sequences, 2020, https://oeis.org/A000170.
[27] Barry Smyth, Paul McClave, Similarity vs. Diversity, in: David W. Aha, Ian D. Watson (Eds.), Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning, ICCBR 2001, Vancouver, BC, Canada, July 30 – August 2, 2001, Proceedings, in: Lecture Notes in Computer Science, vol. 2080, Springer, 2001, pp. 347–361.